

Podstawy Programowania

Laboratorium 6

Wybrane struktury danych

prowadzący: mgr inż. Marta Lampasiak, mgr inż. Michał Jaroszczuk

1 Wprowadzenie

Celem zajęć jest poznanie podstaw programowania w języku C/C++, w szczególności oprogramowanie wybranej dynamicznej struktury danych: listy wskaźnikowej, kolejki, kolejki priorytetowej lub drzewa.

1. Dostęp do danych struktury poznanej na poprzednich zajęciach jest możliwy również za pomocą wskaźnika. Postępowanie jest podobne jak w przypadku zwykłej zmiennej. Należy utworzyć wskaźnik i przypisać mu adres obiektu typu strukturalnego. Przeanalizuj poniższy kod.

```
1 #include <iostream>
2
3 struct Point {
4     double x, y;
5 };
6
7 void print_1(Point *wsk) {
8     std::cout << (*wsk).x << "\n" << (*wsk).y;
9 }
10
11 void print_2(Point *wsk) {
12     std::cout << wsk->x << "\n" << wsk->y;
13 }
14
15 int main() {
16     Point A = {1.5, 2.0};
17     Point* wskPoint = &A;
18
19     print_1(wskPoint);
20     std::cout << "To samo co:" << std::endl;
21     print_2(wskPoint);
22 }
```

Chcąc odnieść się do wartości pól struktury poprzez wskaźnik należałoby zapisać: `(*wskaznik_na_obiekt).pole_struktury`.

Używa się w tej sytuacji nawiasów, ponieważ `.` ma wyższy priorytet niż `*`. Dlatego też zastąpiono przedstawiony zapis na: `->`, a więc:

`wskaznik->pole`.

Jest to o wiele czytelniejsze. Tak więc funkcje `print_1()` i `print_2()` wyświetlą to samo. W ten oto sposób dochodzimy do pojęcia struktury dynamicznej. Analogicznie do tablicy dynamicznej, wykorzystywany jest tutaj operator `new` do dynamicznej alokacji pamięci. Popatrz na przykład poniżej. Zawsze koniecznie pamiętaj o zwolnieniu pamięci.

```
1 #include <iostream>
2 using namespace std;
3
4 struct Point {
5     double x, y;
6 };
```

```

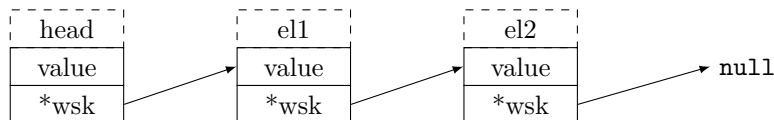
7
8 int main() {
9     Point* A = new Point; // wskaźnik *A na strukturę
10
11     A->x = 1.2;
12     A->y = 1.3;
13     cout << A->x << " " << A->y << endl;
14
15     delete A; // usuwamy obiekt
16     return 0;
17 }

```

2. Na podstawach programowania poznamy 4 podstawowe struktury danych:

- Listę jednokierunkową
- Listę dwukierunkową
- Kolejkę LIFO
- Kolejkę FIFO

Lista jest dynamiczną strukturą danych, w której dane są ułożone w porządku liniowym. Każdy element listy zawiera dwa pola: (1) wartość (2) wskaźnik na kolejny element. Pierwszy element listy zwyczajowo nazywa się głową (*head*), a ostatni ogonem (*tail*). Ostatni element wskazuje na `null`. Aby dostać się do wybranego elementu trzeba przejrzeć całą listę od początku. Jedną z zalet listy jest szybkie usunięcie elementu, ponieważ wystarczy odpiąć odpowiedni element oraz przełączyć element go poprzedzający. W liście dwukierunkowej każdy element oprócz podstawowych pól zawiera dodatkowo wskaźnik na poprzedni element.



Rysunek 1: Graficzna reprezentacja lity

Kolejka jest również dynamiczną strukturą danych. Może zrobić jej implementację przy użyciu listy jednokierunkowej. Jednak w kolejce mamy ograniczony dostęp danych w zależności od typu kolejki jaki wybieramy:

- FIFO, First In, First Out,
- LIFO, last In, First Out.

Każdy nowy element zawsze dodajemy na końcu. Pobrać możemy jedynie pierwszy lub ostatni element. Nie mamy dostępu do elementów wewnątrz kolejki, ale znamy ich liczbę. W kolejce priorytetowej elementy są ułożone według zadanego z góry priorytetu.

2 Zadanie

Stwórz prosty system bazodanowy. Na samym początku wybierz, jaką ocenę chcesz uzyskać. Co oznacza, że wybierając zadanie na ocenę 4.0, nie wykonujesz już zadania na 3.0, ale nie masz szansy na ocenę 5.0. Do każdej implementacji należy napisać takie funkcje, jak:

- dodawanie elementu na sam koniec,
- wyświetlenie wszystkich elementów,
- wyświetlanie elementu o danym indeksie.

Warunkiem zaliczenia ćwiczenia jest wykorzystanie wskaźników i operatora -> do poruszania się po wybranej strukturze. Program powinien mieć formę MENU użytkownika.

1. OCENA 3:

Napisz implementację tablicy dynamicznej. Rozmiar tablicy powinien być równy ilości elementów, które się w niej znajdują i zwiększany przy każdym dodaniu nowego elementu.

2. OCENA 4:

Napisz implementację listy jednokierunkowej (L).

3. OCENA 5:

Napisz implementację kolejki FIFO przy użyciu listy jednokierunkowej lub dwukierunkowej.

Przykładowe struktury danych:

- student
 - nr indeksu,
 - imię,
 - nazwisko,
 - płeć,
 - data urodzenia,
- samochód
 - nr rejestracyjny,
 - marka,
 - model,
 - typ,
 - rocznik,
 - moc.

W kodzie pisanych programów proszę umieszczać komentarze. Brak komentarzy uniemożliwi zdobycie maksymalnej ilości punktów za zadania!

Uwaga

Przypominam o przesłaniu programów na koniec zajęć, według wcześniej podanego wzoru. Proszę o niedołączanie plików innych niż o rozszerzeniu *.cpp.